Independent degree project - first cycle

Bachelor's thesis – 15 ECTS credits

Master of Science in Engineering:

Industrial Engineering and Management

Data Mining for Network Intrusion Detection

A comparison of data mining algorithms and an analysis of relevant features for detecting cyber-attacks

Rebecca Petersen

Mittuniversitetet

MID SWEDEN UNIVERSITY

Campus Härnösand Universitetsbacken 1, SE-871 88. **Campus Sundsvall** Holmgatan 10, SE-851 70 Sundsvall.
Campus Östersund Kunskapens väg 8, SE-831 25 Östersund.
Phone: +46 (0)771 97 50 00, Fax: +46 (0)771 97 50 01.

MID SWEDEN UNIVERSITY
Department of Information and Communication Systems (IKS)

Examiner: Aron Larsson, aron.larsson@miun.se
Internal Supervisor: Leif Olsson, leif.olsson@miun.se
External Supervisor: Ross Tsagalidis, Swedish Armed Forces
External Supervisor: Ingvar Ståhl, Swedish Armed Forces
Author: Rebecca Petersen, repe1000@student.miun.se
Degree programme: Master of Science in Engineering: Industrial Engineering and Management, 300 higher credits
Semester, year: Summer, 2015.

SWEDISH ARMED FORCES

Mittuniversitetet
MID SWEDEN UNIVERSITY

Abstract

Data mining can be defined as the extraction of implicit, previously unknown, and potentially useful information from data. Numerous researchers have been developing security technology and exploring new methods to detect cyber-attacks with the DARPA 1998 dataset for Intrusion Detection and the modified versions of this dataset KDDCup99 and NSL-KDD, but until now no one have examined the performance of the Top 10 data mining algorithms selected by experts in data mining. The compared classification learning algorithms in this thesis are: C4.5, CART, k-NN and Naïve Bayes. The performance of these algorithms are compared with accuracy, error rate and average cost on modified versions of NSL-KDD train and test dataset where the instances are classified into normal and four cyber-attack categories: DoS, Probing, R2L and U2R. Additionally the most important features to detect cyber-attacks in all categories and in each category are evaluated with Weka's Attribute Evaluator and ranked according to Information Gain. The results show that the classification algorithm with best performance on the dataset is the k-NN algorithm. The most important features to detect cyber-attacks are basic features such as the number of seconds of a network connection, the protocol used for the connection, the network service used, normal or error status of the connection and the number of data bytes sent. The most important features to detect DoS, Probing and R2L attacks are basic features and the least important features are content features. Unlike U2R attacks, where the content features are the most important features to detect attacks.

Keywords: data mining, machine learning, cyber-attack, NSL-KDD, features, DoS, Probing, R2L, U2R.

Acknowledgements

This bachelor thesis has been created during the summer of 2015 as a part of the Master of Science in Engineering program with a major in Industrial Engineering and Management at Mid Sweden University. First of all, I would like to thank my external supervisor Ross Tsagalidis at the Swedish Armed Forces, for all the support and enthusiasm during this summer.

I would like to thank my external supervisor Ingvar Ståhl at the Swedish Armed Forces, for valuable comments that have helped me to consider new perspectives. I would like to express my appreciation to my supervisor at the Mid Sweden University, Leif Olsson, for the generosity and guidance during our meetings. I would also like to thank Lennart Franked, at the Mid Sweden University, for the inspiration and assistance regarding network security and cyber-attacks.

Last but not least, I would like to thank my family, especially David, for the patience, understanding and enormous support that have helped me to fulfil my dreams.

Table of Contents

Terminology

Acronyms and Abbreviations

AFRL	Air Force Research Laboratory
ANN	Artificial Neural Networks
ARFF	Attribute-Relation File Format
BIOS	Basic Input/Output System
CART	Classification and Regression Trees
DARPA	Defence Advanced Research Projects Agency
DoS	Denial-of-Service attack
IDS	Intrusion Detection System
ICMP	Internet Control Message Protocol
KDD	Knowledge Discovery in Databases
k-NN	k-Nearest Neighbour
MIT	Massachusetts Institute of Technology
NSL	Network Security Laboratory
Probe	Probing attack
R2L	Remote-to-Local attack
SVM	Support Vector Machines
SQLi	Structured Query Language injections
TCP	Transmission Control Protocol
TXT	Text
U2R	User-to-Root attack
UDP	User Datagram Protocol
XSS	Cross-Site Scripting

1 Introduction

1.1 Background and problem motivation

For generations humans have protected their privacy and property with locks, fences, signatures and seals with support by national laws, manners and customs. Almost every systems is today automated and electronic, with control over information systems hackers can for example falsify communications to aircraft, freeze bank assets, manipulate traffic lights, delete satellite data and shut down military control systems. One of the most challenging and important tasks for engineers in the twenty-first-century is to build security technology to detect cyber-attacks and network intrusions in order to protect privacy and property. [1]

A signature based Intrusion Detection System (IDS) is used to search for network traffic known to be malicious. These IDSs depend on manually created intrusion patterns and if the signature based IDS is not updated, cyber-attacks are not being noticed [2]. The gap between generation of data and our understanding of it is growing. As the volume of data increases, the proportion of it that people understand is alarmingly decreasing. Data mining can be defined as the extraction of implicit, previously unknown, and potentially useful information from data. [3]

The Defence Advanced Research Projects Agency (DARPA) 1998 dataset for Intrusion Detection Evaluation and the modified versions of this dataset, Knowledge Discovery and Data mining Competition (KDDCup) 1999 dataset and NSL-KDD dataset, is widely used to evaluate performance of Intrusion Detection Systems (IDS). These datasets consist of simulated network traffic created to be comparable to the system at an American Air Force base with additional cyber-attacks [4]. Numerous researchers have been developing security technology and exploring new methods to detect cyber-attacks with these datasets, but until now no one have examined the performance of the Top 10 data mining algorithms [5] selected by experts in data mining.

To upgrade security configuration of a network it is important to know how to detect various cyber-attacks. Further research to expand the knowledge about data mining for network intrusion detection is necessary to develop new security technology.

1.2 Overall aim

The overall aim is to increase the knowledge about data mining for intrusion detection, in order to detect cyber-attacks and make well-founded decisions about defence configuration.

1.3 Scope

The scope of this thesis is a comparison of the classification algorithms present in the raked list *Top 10 algorithms in data mining* [5]: C4.5, Classification and Regression Trees (CART), k-Nearest Neighbour (k-NN) and Naïve Bayes. No other machine learning algorithms are taken into account. The performance of the selected machine learning algorithms are evaluated with accuracy, error rate and average cost. This thesis does not include an analysis if the selected machine learning algorithms perform differently in other environment, for example performance on other datasets from military network traffic.

The study is delimited to the dataset DARPA 1998 dataset for Intrusion Detection Evaluation presented in Chapter 2.4. This dataset includes simulated cyber-attacks in the following categories: Denial of Service (DoS) attack, Remote to Local (R2L) attack, User to Root (U2R) attack and Probing attack. No other cyber-attacks than those presented in Chapter 2.4 have been taken into account.

To know what to do when early detecting an anomaly or cyber-attack is vital. This sort of risk assessment is a very interesting topic, but this is not included in this thesis.

1.4 Concrete and verifiable goals

In order to analyse how data mining can be used to detect network intrusions and cyber-attacks, four goals have been set:

- Evaluate which classification machine learning algorithm that is most accurate in detecting intrusions.

- Explore if the number of classes used for classification in the dataset is vital.

- Examine which features that are most important to analyse to detect any anomaly.

- Examine which features that are most important to analyse to detect intrusions in each of the following categories: DoS, Probing, U2R and R2L.

1.5 Outline

The first chapter gives the reader an introduction and basic understanding for this thesis. In Chapter 2 theory and related work is presented. The methodology is described in Chapter 3, this Chapter also includes motivations for the chosen method and a method discussion. All results are presented in Chapter 4 and a discussion of the obtained results is available in Chapter 5. Finally, Chapter 6 presents conclusions and suggestions for future research.

2 Theory and related work

This Chapter presents theory of cyber-attacks, data mining, selected machine learning algorithms, the data mining tool Weka and the selected dataset DARPA 1998 dataset for Intrusion Detection Evaluation together with previous work related to this thesis.

2.1 Cyber-attacks

The range of hostile activities that can be performed over information networks are enormous, from vandalism of websites to large-scale damage to military or civilian infrastructures that depend on those networks. Hence, definitions of the term *cyber-attack* vary. A cyber-attack can be defined by efforts to alter, disrupt, or destroy computer systems or networks or the information or programs on them. [6]

Cyber-attacks are continuously evolving and becoming more complex and frequent. Some experts in intrusion detection argue that most novel cyber-attacks are variants of known attacks and that the signature of known attacks can be enough to detect novel attacks. [7]

Four categories of cyber-attacks are presented by Tavallaee, Bagheri, Lu and Ghorbani [8] as follows:

* *Denial of Service Attack (DoS)* are attacks in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

* *User to Root Attack (U2R)* are attacks where the attacker gain access to a normal users account on the system, perhaps by social engineering, hacking or sniffing passwords. Then the attacker takes advantage of some weakness to gain root access to the system.

* *Remote to Local Attack (R2L)* are attacks where the remote attacker tries to gain access to a local user account.

* *Probing attack* are attacks where the attacker tries to gain information about a network of computers to avoid its security control.

Predictably, attackers are learning how to avoid countermeasures. The attackers have the advantage to plan strategically and choose the best time to attack and find vulnerabilities in networks. Nations and organizations need knowledge to defend themselves against cyber-attacks. Intrusion detection needs to get a whole lot better in order to detect cyber-attacks and to make the best decisions for defence configuration. [9]

2.2 Data Mining

Today's society feeds on information produced by digital communication and computing. Most of the information is in its raw form: data. If *data* is defined as recorded facts, then *information* is the set of patterns that underlie the data. The gap between generation of data and our understanding of it is growing. As the volume of data increases, the proportion of it that people understand is decreasing. Hidden in this large amount of data is information, potentially useful information, which unfortunately seldom is taken advantage of. [3]

In science, finance, marketing, health care, retail, or any other field, the traditional approach to turn data into information relies on manual analysis and interpretation with one or more analysts. As data volumes grow dramatically, this manual analysis is expensive, slow and subjective. The concept of finding useful information in data has been given different names, for example, data mining, knowledge extraction, data archaeology and data pattern processing. The phrase Knowledge Discovery in Databases (KDD) was invented at a workshop in 1989. If KDD is the overall process of finding useful knowledge from data including how the data is stored and accessed; then *data mining* is a particular step in this process. Data mining is the application of specific algorithms for extracting information from data. The term data mining is historically used mostly by statisticians and data analysts, but has gained popularity in other fields during recent years. [10]

Data mining can be used in many areas, for example web mining, decision involving judgement, screening images, load forecasting, medical diagnosis, marketing, sales and manufacturing. The technical basis of data mining is machine learning methods. These are used to find and extract information from raw data. In data mining, the data is stored electronically and the process of finding patterns in data is automatic or semiautomatic. [3]

The input to a machine learning algorithm is a set of *instances*. An instance is an individual, independent example of the concept we want to learn. Each instance consists of predetermined *features* (also called *attributes*) whose values measure different aspects of the instance. The features can be numerical (also called continuous) or nominal (also called discrete). Represented in a matrix, the instances are the rows and the features are the columns. For example, if the instances are land vehicles, then "number of wheels" and "colour" are features. [3]

Four different types of machine learning appear in data mining applications: classification learning, association learning, clustering and numeric prediction. *Classification learning* consists of presenting the learning scheme with a set of classified instances and from this learns how to classify unseen instances. *Association learning* aims to map any association between features. *Clustering* form groups of instances that belong together. *Numeric prediction* is a variant of classification learning where the predicted outcome is a numeric quantity instead of a discrete class. [3]

In classification learning the success can be measured by giving the algorithm a training dataset and an independent test dataset in which the true classifications are known but not available to the machine. The success rate is an objective measure of how well the machine learning algorithm can classify unknown instances. [3]

The following subchapters present four machine learning algorithms used in classification learning.

2.2.1 C4.5

Wu et al. [5] describes the C4.5 algorithm which first starts with building a decision tree using the divide-and-conquer algorithm as follows:

- Given a set of *S* instances. If all the instances in *S* belong to the same class or *S* is small, the tree is a leaf labelled with the most frequent class in *S*.
- Otherwise, choose a test based on a single feature with two or more outcomes. Make this test the root of the tree with one branch for each outcome of the test, partition *S* into corresponding subsets S_1,S_2,… according to the outcome for each instance, and apply the same procedure recursively to each subset.

The test chosen in the second step of the divide-and-conquer algorithm above are information gain and information gain ratio for the C4.5 algorithm [5]. According to [3], [11] the information gain is calculated as follows:

Let S be a set of training instances with their corresponding labels. Suppose there are m classes and the training set contains s_i instances of class i. The expected information gain needed to classify a given instance is calculated by:

$$I(s_1, s_2, \dots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \, log_2(\frac{s_i}{s}) \tag{1}$$

A feature F with values $\{f_1, f_2, \dots, f_v\}$ can divide the training set into v subsets $\{S_1, S_2, \dots, S_v\}$ where S_j is the subset which has the value f_j for feature F. Let S_j contain s_{ij} instances of class i. Entropy of the feature F is:

$$E(F) = \textstyle\sum_{j=1}^{v} \frac{s_{1j}+\dots+s_{mj}}{s} \times I(s_{1j}, \dots, s_{mj}) \tag{2}$$

Information gain, $G(F)$, for feature F can be calculated as:

$$G(F) = I(s_1, s_2, \dots, s_m) - E(F) \tag{3}$$

The intrinsic information $I_{Int}(F)$ of a Feature F is:

$$I_{Int}(F) = -\sum_{i=1}^{m} \frac{|s_i|}{|s|} \, log_2(\frac{|s_i|}{|s|}) \tag{4}$$

The Gain Ratio is a modification of the information gain to reduce its bias towards multi-valued features. The Gain Ratio is simply the information gain divided by the intrinsic information:

$$GR(F) = \frac{G(F)}{I_{Int}(F)} \tag{5}$$

Further, if the dataset is too complex the phenomenon overfitting can occur. Overfitting is when the decision tree describes random error instead of the underlying relationship, which can lead to good performance in training set but poor performance on the test dataset. [3]

To avoid overfitting the created decision tree is pruned from the leaves to the root. The C4.5 algorithm uses a single-pass algorithm based on an estimate of the pessimistic error rate associated with a set of N instances and E instances of which do not belong to the most frequent class. C4.5 then defines the upper limit of the binomial probability when E instances have been observed in N trials, using a user-specified confidence value whose default value is 0.25. To drop conditions and achieve the lowest pessimistic error rate, a hill-climbing algorithm is used. The final rule set to the decision tree which is used to classify instances is then simplified and far fewer than the number of leaves on the pruned decision tree. The C4.5 algorithm main disadvantage is the amount of CPU time and memory required. [5]

2.2.2 CART

CART is an abbreviation of Classification and Regression Trees. The decision tree in CART is a binary recursive procedure. Compared with the C4.5 algorithm, the CART algorithm creates a sequence of trees instead of one. [5]

The trees in CART are created by the same divide-and-conquer algorithm presented in Chapter 2.2.1 on C4.5, but in the second step where C4.5 use an information-based test, the CART use the Gini impurity index to rank tests. The Gini impurity index is calculated by summarizing the probability of each instance being chosen multiplied by the probability of a mistake in classifying that instance. The Gini impurity index is zero when all instances in the node are classified as the same category. [5]

The Gini impurity index, $G_I(t)$, of a node t in a binary decision tree is calculated by:

$$G_I(t) = 1 - p(t)^2 - (1 - p(t))^2 \qquad (6)$$

Where $p(t)$ is the (possibly weighted) relative frequency of the first class in the node. The gain generated by a split of the parent node P into left and right children L and R is:

$$I_G(P) = G_I(P) - qG_I(L) - (1 - q)G_I(R) \qquad (7)$$

Where q is the (possibly weighted) fractions of instances going left. [5]

The trees are pruned with cost-complexity pruning and the most optimal tree is identified by evaluating the predictive performance in the pruning sequence [5]. The idea of cost-complexity pruning is to first prune the subtrees that, relative to their size, lead to the lowest increase in error on the training data. The quantity α measure the increase in average error per leaf in the concerned subtree. A sequence of smaller pruned trees is generated by monitoring α in the pruning process. All subtrees with the smallest value of α, among the current version of the tree, in each iteration, are pruned. Each candidate tree in the resulting set of pruned trees has a particular threshold value α_i. The optimal tree can be chosen by using a test dataset to estimate the error rate of each tree. [3]

2.2.3 K-Nearest Neighbour

The k-Nearest Neighbour (k-NN) is an instance based learning method. In instance learning the algorithm store instances from the train dataset and compare unknown instances in the test dataset with those. The k-NN algorithm is one of the most simple machine learning algorithms. This algorithm is a type of lazy learning where the instances initially are approximated locally and all computations occurs at classification. The k-NN algorithm uses all classified training instances to determine a local hypothesis function. Then the test dataset are compared to the stored training instances and each instance are assigned to the same class as the k most similar stored instances. If k=1, then an instance is assigned to the class of the most similar instances. [3]

To compare the unknown instances in the test dataset and the known instances stored from the train dataset the k-NN algorithm uses a distance metric. The computation of distance between two or several numeric features is trivial. In general the standard Euclidian distance is used. When nominal features are present in the dataset the distance between different values are zero if the values are identical, otherwise the distance is one. [3]

Wu et al. [5] describe the output of the k-NN classification algorithm as follows. Given a training dataset D and a test instance $z = (\mathbf{x}', y')$. Where $\mathbf{x}$ is the data of a training instance and y is its class. Likewise, $\mathbf{x'}$ is the data of a test instance and y' is its class.

Further, the authors describe that the algorithm computes the distance (or similarity) between z and all the training instances $(\mathbf{x}, y) \in D$ to determine a list of its nearest neighbours D_Z. The test instance is classified based on the majority class of its nearest neighbours obtained from the list D_Z:

$$\text{Majority Voting: } y' = \underset{v}{argmax} \quad \sum_{(x_i, y_i) \in D_z} I(v = y_i), \qquad (8)$$

where v is a class label for the ith nearest neighbours, and $I(\cdot)$ is an indicator function that returns 1 if the argument is true and 0 otherwise. [5]

The k-nearest neighbour classification algorithm is described Wu et al. [5] as follows:

- Input: *D*, the set of *k* training instances, and test instance $z = (\mathbf{x}', y')$.
- Process: Compute $d(\mathbf{x}', \mathbf{x})$, the distance between z and every instance, $(\mathbf{x}, y) \in D$. Select $D_Z \subseteq D$, the set of *k* training instances to *z*.
- Output: $y' = \underset{v}{argmax} \quad \sum_{(x_i, y_i) \in D_z} I(v = y_i)$.

The disadvantage of storing all instances from the train dataset in memory is slow calculations and consumption of large amounts of storage [3]. The k-NN algorithm can be expensive for large training datasets because of the requirements of computing distance of a test instance to all the instances in the labelled train dataset. Classification with k-NN is particularly well suited for multi-modal classes. [5]

2.2.4 Naïve Bayes

The Naïve Bayes are simple to apply to huge data sets due to properties as easy to construct and no need for any complicated iterative parameter estimation scheme [5]. Wu et al. [5] describes the basic principle of the Naïve Bayes algorithm with only two classes labelled $i = \{0, 1\}$ to simplify presentation of the algorithm. The aim is to use the training set with corresponding labels to construct a score such that smaller scores are related to class 0 and larger scores with class 1. To classify instances this score is compared to a threshold, *t*. If $P(i|x)$ is defined as the probability that an instance with measurement vector $x=(x_1,\ldots,x_p)$ belongs to class *i*, and $P(i)$ is the probability that an instance will belong to class *i* if we know nothing more about it and $f(x|i)$ is the conditional distribution of *x* for class *i* instances. Then $P(i|x)$ are proportional to $f(x|i)P(i)$, and the following ratio are a suiTable score:

$$\frac{P(1|x)}{P(0|x)} = \frac{f(x|1)P(1)}{f(x|0)P(0)} \tag{9}$$

According to Wu et al. [5] the Naïve Bayes is assuming that the training set are a random sample from the overall population, the $P(i)$ can be estimated from the proportion of class *i* instances in the training set. To estimate *f(x|i)* the Naïve Bayes assumes that the features of *x* are independent:

$$f(x|i) = \prod_{j=1}^{p} f(x_j|i) \tag{10}$$

and then separately estimates each of the univariate distributions $f(x_j|i)$, *j = 1,..,p; i=0,1.* Now the *p* dimensional multivariate problem is reduced to *p* univariate estimation problems. Given the independence in (10), the ratio in (9) can be written as follows [5]:

$$\frac{P(1|x)}{P(0|x)} = \frac{\prod_{j=1}^{p} f(x_j|1)\, P(1)}{\prod_{j=1}^{p} f(x_j|0)\, P(0)} = \frac{P(1)}{P(0)} \prod_{j=1}^{p} \frac{f(x_j|1)}{f(x_j|0)} \tag{11}$$

An alternative score, with also is a monotonic function which can be related to *P(i|x)* is the logarithm. [5]

$$ln\frac{P(1|x)}{P(0|x)} = ln\,\frac{P(1)}{P(0)} + \sum_{j=1}^{p} ln\,\frac{f(x_j|1)}{f(x_j|0)} \tag{12}$$

This sum can be written as:

$$ln\frac{P(1|x)}{P(0|x)} = k + \sum_{j=1}^{p} w_j \tag{13}$$

Where w_j and *k* is defined by:

$$w_j = ln\,\frac{f(x_j|1)}{f(x_j|0)} \tag{14}$$

$$k = ln\,\frac{P(1)}{P(0)} \tag{15}$$

The sum in (13) shows the simple structure of the Naïve Bayes. The Naïve Bayes algorithms is one of the oldest formal classification algorithms and possess qualities such as simplicity, elegance and robustness.[5]

2.2.5 Evaluation of machine learning algorithms

When comparing different algorithms based on their performance on the dataset, the rankings will be different depending on the chosen performance measures. It is important that the performance is measured on one or a few of the most relevant aspects of the dataset. [12]

To evaluate different machine learning algorithms for classification problems the error rate on the test set will give a good indication on future performance of the algorithm. The algorithm predicts the class of each instance, and if it is correct, that is counted as a success; otherwise it is an error. The error rate measures the overall performance of the classifier on the test dataset. [3]

The most natural performance measure on instances on a single class is the true-positive and false-positive rate. The true-positive rate refers to the proportion of the instances of some class *i* of interest actually assigned to class *i* by the learning algorithm. The false-positive rate refers to the proportion of the instances of some class *i* of interest actually assigned incorrectly to class *i* by the learning algorithm.[12]

Performance measures such as error rate, true-positive and false-positive rates does not take cost into account. If the costs are known they can be used in a financial analysis. In a multiclass problem, the cost matrix is a square matrix whose size is the number of classes, the diagonal elements represent the cost of correct classification. The average cost per decision can be calculated as follows: [3]

$$\text{Average cost} = \frac{\sum(M_{ij} \times C_{ij})}{S} \qquad (16)$$

where M_{ij} is the number of instances of class *i* incorrectly classified as class *j*, C_{ij} is the corresponding cost from the cost matrix and S is the total number of instances in test dataset. [3]

2.2.6 Feature Evaluation

There are two different approaches when selecting a good feature subset. One is to evaluate using a machine learning algorithm, the other is to make an independent assessment based on general characteristics of the data. The first approach is called the *wrapper* method due to wrapping the learning algorithm into the feature selection. The second approach is called the *filter* method because the data is filtered before applying a learning algorithm. [3]

One strategy to rank the feature with the filter method is using their Information Gain (presented in Chapter 2.2.1 about C4.5). The features are then ranked by measuring the information gain with respect to the class. [3]

2.3 Weka

The Weka workbench is a open source software providing a collection of machine learning algorithms and data pre-processing tools. Weka is available online at the website of the Machine Learning Group at the University of Waikato in New Zealand [13].

Weka is designed for quickly trying out existing data mining methods on any dataset. The system is developed at the University of Waikato and is distributed under the conditions of the GNU General Public License for Linux, Windows and Macintosh. [3]

Weka has become one of the most widely used machine learning tool. Weka is continually growing and includes methods for the main data mining problems such as: feature selection, association rule mining, classification, clustering and regression. All algorithms require their input in the form of a single relational table in the Attribute-Relation File Format (ARFF). Weka can be used to apply a learning method to a dataset headed for learning more about the data or to apply several different learning methods to compare their performance in order to choose one for prediction. [3]

2.4 DARPA 1998 Dataset for Intrusion Detection Evaluation

The Lincoln Laboratory at Massachusetts Institute of Technology (MIT) university was sponsored by Defence Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), to create the first standard corpora for computer network Intrusion Detection Systems (IDS). The dataset is named *DARPA Dataset for Intrusion Detection Evaluation* and can be used for training and evaluating different IDSs. These evaluations contributed considerably to the research regarding intrusion detection due to the objectivity, formality, repeatability and statistically significance. The dataset interest researchers who work on general problems of workstation and network intrusion detection. [4]

The DARPA dataset consists of network traffic similar to the actual traffic between a small Air Force base and the internet. The main reason for not using operational traffic at an Air Force base and controlled live attacks were the fact that it would release private information, compromise security and might damage military systems. To ensure that evaluation of an IDS with the dataset could uncover weaknesses in many different IDSs, widely varied cyber-attacks were developed which might be used by both highly skilled and novice attackers. The four categories of attacks included in the dataset are DoS, R2L, U2R and Probing. [14]

The variability of traffic over time and the overall proportion of traffic from different services are similar to what's observed on Air Force Bases. All attacks are launched from the outside of the simulated base and evidence of each attack is captured by a sniffer positioned at the entrance of the network. [14]

The DARPA 1998 Dataset for Intrusion Detection Evaluation consists of two sets of data: seven weeks of training data containing normal background traffic and labelled attacks, and two weeks of test data with unlabelled data including new attacks [14]. In Table 1 and Table 2 below the attacks in train and test datasets are presented. A more detailed description of each attack presented in Table 1 and Table 2 is available in at the Lincoln Laboratory's Attacks database [4].

Table 1: The 22 attacks in DARPA 1998 train dataset for Intrusion Detection Evaluation[4].

Attack category	**Attack**
DoS	Back, Land, Neptune, Pod, Smurf, Teardrop.
Probing	Satan, Ipsweep, Nmap, Portsweep.
R2L	Guess_Passwd, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy.
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl.

Table 2: The 37 attacks in DARPA 1998 test dataset for Intrusion Detection Evaluation [4].

Attack category	**Attack**
DoS	Back, Land, Neptune, Pod, Smurf, Teardrop, Mailbomb, ProcessTable, Udpstorm, Apache2, Worm.
Probing	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint.
R2L	Guess_Passwd, Ftp_write, Imap, Phf, Multihop, Warezmaster, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named.
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps.

The DARPA 1998 dataset consists of instances with information about network connections. Each instance consists of 41 features. The features (also called features) are grouped into four categories: *Basic features* used for general-purpose traffic analysis. *Content features* seek suspicious behaviour e.g. number of failed login attempts. [15]

Time-based traffic features consists of "same host" features that inspect the connections in the past 2 seconds that have the same destination host as the current connection. This category also includes features of "same service" that inspect connections in the past 2 seconds that have the same service as the current connection. Detecting attacks that use a larger time interval than 2 seconds is done by *host-based traffic* features which consists of 100 connections and corresponds to the time-based traffic features. [15]

In Table 3 below all 41 features are presented, for more extensive description see Appendix A. Feature number 1-9 belongs to the category Basic Features, feature number 10-22 belongs to the category Content Features, feature number 23-30 belongs to the category Time-based Traffic Features and feature number 31-41 belongs to the category Host-based Traffic Features.

Table 3: All features in DARPA 1998 dataset for Intrusion Detection Evaluation [15].

No.	Feature name	No.	Feature name
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_host_rate
11	num_failed_logins	32	dst_host_count
12	logged_in	33	dst_host_srv_count
13	num_compromised	34	dst_host_same_srv_ rate
14	root_shell	35	dst_host_diff_srv_rate
15	su_attempted	36	dst_host_same_src_port_rate
16	num_root	37	dst_host_srv_diff_host_rate
17	num_file_creations	38	dst_host_serror_rate
18	num_shells	39	dst_host_srv_serror_rate
19	num_access_files	40	dst_host_rerror_rate
20	num_outbound_cmds	41	dst_host_srv_rerror_rate
21	is_hot_login		

Thomas, Sharma and Balakrishnan [7] studied the usefulness of DARPA dataset in 2008, 10 years after the creation of the dataset. The authors' conclusion is that the dataset is not obsolete and that the dataset can still be used to evaluate IDSs. The DARPA dataset have proved to have the vital potential in common attacks on the network traffic such as DoS, R2L, U2R and Probing. Finally the authors note that if an IDS is evaluated on the DARPA dataset, this indicates its performance on the real network traffic. According to Thomas, Sharma and Balakrishnan the dataset can be considered as the base line of any research.

McHugh [16] criticise the dataset on the procedures used when building the dataset such as the use of synthetic simulated data due to concerns of privacy and the sensitivity of actual intrusion data. McHugh claims that the models used to generate background traffic in the DARPA dataset are too simple and criticises the choice to implement attacks via scripts and a varied set of programs.

Mahoney and Chan [17] analysed the DARPA dataset from 1999 which have several improvements from previous dataset. The authors similarly comment that the background traffic in the DARPA dataset differs a lot from real background traffic mainly because of the large and growing range for some features in real traffic and the fixed range of the same features in the DARPA dataset. This generates false alarms in many IDSs. Further Mahoney and Chan state that if an IDS can not perform well in detecting the attacks in the DARPA dataset, it can not perform acceptably on a real data.

2.4.1 KDD Cup 1999 Dataset

The dataset KDD Cup 1999 is the most widely used dataset for evaluation of signature-based IDSs since 1999. The KDD Cup '99 dataset is based on the DARPA 1998 Dataset for Intrusion Detection Evaluation and was created for the third International Knowledge Discovery and Data Mining Tools Competition. [8]

The KDD Cup'99 training data set consists of approximately 4,900,000 single connections each of which contains 41 features and is labelled as either normal or an attack, with one specific attack type [8].

The cost matrix used for scoring in the KDD Cup'99 competition is presented below in Table 4. The columns correspond to predicted categories and the rows correspond to actual categories. [18]

Table 4: Cost matrix for KDD Cup'99 dataset.[18]

	Normal	DoS	Probe	R2L	U2R
Normal	0	2	1	2	2
DoS	2	0	1	2	2
Probe	1	2	0	2	2
R2L	4	2	2	0	2
U2R	3	2	2	2	0

Tavallaee, Bagheri, Lu and Ghorbani [8] present two different approaches which is used by researchers to apply the KDD dataset. In the first approach, a dataset is created using sampling of both train and test sets of the KDD Cup'99 dataset. In the second approach, the training set is created using sampling of the KDD Cup'99 train dataset and the test set are arbitrarily selected from the KDD Cup'99 test dataset.

The first approach is used by [19], [20], [21], [22], [23], [24] where the used dataset is sampled randomly from KDD Cup'99 dataset, corresponding to 10% of the original dataset. The creation of training and test dataset with sampling gives all researchers' different datasets, consequently the results will be inconsistent and incomparable [8].

2.4.2 NSL-KDD dataset

Tavallaee et al. [8] conducted a statistical analysis of the KDD Cup'99 dataset. The authors found a large number of redundant records, 78% in the train dataset and 75% in the test dataset. This amount of duplicated records can prevent data mining algorithms to learn from infrequent records such as U2R attacks. The authors also note that the duplicated records in the KDD Cup'99 test dataset will cause the evaluation results to be biased by the algorithms with better detection rates on frequent records.

Tavallaee et al. [8] created the NSL-KDD dataset in 2009 from the KDD Cup'99 dataset to solve the issues mentioned above, by removing redundant records. The NSL-KDD train dataset consists of 125,973 records and the test dataset of 22,544 records, compared to the KDD Cup'99 train dataset of 4,900,000 records and the test dataset of about 2,000,000 records. The authors argue that the size of the NSL-KDD dataset is reasonable which makes it affordable to use the complete NSL-KDD dataset without the need to sample randomly. This gives consistent and comparable results of different research works.

2.5 Related work

An analysis of 10% of the KDD Cup'99 training dataset using clustering with the k-means clustering algorithm and Oracle 10g data miner is performed by [19]. Their aim was to establish a relationship between different attacks and the protocol used (TCP, UDP and ICMP). The result was that TCP is affected by 19 of 22 attacks.

10% of KDD Cup'99 dataset is used in [20] with k-means clustering algorithm in Weka. A method of 7 steps is proposed to achieve high accuracy and low false-positive rate. The result was an accuracy of 9.2013% and a false-positive rate of 97% in the authors' first implementation.

Stratified weighted sampling techniques is used in [21] to generate samples from 10% of the KDD Cup'99 dataset. The authors propose a new decision tree algorithm implemented in MATLAB, which is compared to ID3. The result is that their proposed algorithm has a smaller error rate than ID3.

To identify the most relevant features for attack classification is important to improve intrusion detection according to [22]. The authors used 10% of KDD Cup'99 training data and the test data set to analyse the 41 features with the OneR machine learning algorithm. The performance measured by true-positive and false-positive rates. The results was that the normal class and the attack classes Neptune and Smurf are highly related to certain features. The authors comment that these tree classes make up 98% of their training data, which results in very good classification results with the machine learning algorithm OneR.

Two of the most popular ensemble classifiers Bagging and Boosting are compared to the C4.5 algorithm by [23]. The study is conducted on 10% of the KDD Cup'99 data with Weka. The performance measure used to evaluate the different algorithms are error rate, false-positive and false-negative rates. The results show that the Boosting algorithm AdaBoostM1 gives a significant lower error rate than the other two algorithms.

The whole NSL-KDD dataset is studied by [25] in a comparative study of selected data mining algorithms in Weka. Five popular algorithms are compared: SVM, ANN, k-NN, Naïve Bayes and C4.5. Some of the used performance measure is accuracy, error rate and classification time. The authors result is that C4.5 is best suited for real-time classification due to the high accuracy and relatively fast classification time.

Feature selection relevance analysis is conducted by [11] on 10% of KDD Cup'99 train dataset. The authors calculate information gain for each class (normal or an attack) to analyse all features in the dataset. Due to the unrealistic simplicity of the 10% of KDD Cup'99 train dataset the authors state that the results can be questioned. Their result is that 8 features have a very small maximum information gain (smaller than 0.001). The authors conclude that feature number 9, 20 and 21 (same numbering as Table 3) have no role in detection of any attack. Also, feature number 15, 17, 19, 32 and 40 have minimum role in detection of an attack.

Although the KDD Cup'99 dataset is analysed by many researchers, their approach is often to sample 10% of the train dataset due to the computation cost. This method gives researchers different datasets, and the results will therefore be inconsistent and incomparable. Further research should be conducted on the improved NSL-KDD dataset to get comparable results. Knowledge about the most relevant features for both the whole dataset and each attack category (DoS, Probe, U2R, R2L) in a dataset based on the DARPA 1998 dataset seems to be deficient.

3 Methodology

3.1 Data collection

The data collection consists of gathering both primary data and secondary data. Primary data are the results from Weka and secondary data are literature and research articles, which is used for exploring previous knowledge within the area. Literature and research articles are collected by searching in databases such as ScienceDirect®, IEEE Xplore®, Primo® and Google Schoolar®.

3.2 Choice of dataset

The DARPA Dataset for Intrusion Detection Evaluation is the most used dataset since 1998 for research on intrusion detection, also this dataset includes a wide range of attacks in four main categories [8]. The dataset is created to be comparable to a military system at an Air Force base, which gives the dataset applicability in this thesis. The choice of this particular dataset is due the fact that this dataset will not release private information, compromise security or damage network systems. Further, the DARPA dataset is available online for free which gives the dataset another advantage compared to create a new dataset with network connections.

The improved dataset NSL-KDD is selected due to the possibility to use the complete NSL-KDD dataset, resulting in consistent and comparable results which contributes to research. The creation of a modified dataset with five classes gives the ability to analyse the four different attack categories in a satisfactory way to ensure distinct results. NSL-KDD is already divided into a training dataset and a test dataset, this guarantee the same results if the procedure is repeated.

3.3 Choice of data mining software

The Weka workbench is the most used data mining software to compare different data mining algorithms for intrusion detection with KDD Cup'99 and NSL-KDD. Therefore Weka is chosen in this thesis. An advantage with Weka is that the software is continually developed and easily understood. The default settings for all classification algorithms are carefully selected and well suited for this dataset.

3.4 Choice of data mining algorithms

Witten, Frank and Hall [3] state that the existence of Weka workbench makes it easy to blindly apply data mining algorithms to some dataset. The authors present a Table of the top 10 algorithms in data mining from the 2006 International Data Mining Conference obtained from a survey. The details of the conducted survey are presented by Wu et al. in [5]. The table is presented below in Table 5.

Table 5: Top 10 algorithms in data mining [3][2].

No.	Algorithm	Category
1.	**C4.5**	**Classification**
2	***k*-means**	**Clustering**
3	**SVM**	**Statistical learning**
4.	**Apriori**	**Association analysis**
5.	**EM**	**Statistical learning**
6.	**PageRank**	**Link mining**
7.	**Adaboost**	**Ensemble learning**
8.	**kNN**	**Classification**
9.	**Naïve Bayes**	**Classification**
10.	**CART**	**Classification**

All algorithms which belong to the category *Classification* are chosen: C4.5, k-NN, Naïve Bayes and CART. The selection of classification learning as the selected machine learning scheme is because of the possibility to use a train dataset to learn and to use a test dataset with novel attacks for evaluation. This gives an objective measure of how well the machine algorithm works for intrusion detection. These classification algorithms will give the same result if the procedure is repeated.

3.5 Approach

Two versions of the NSL-KDD train and test datasets was downloaded from the website of the University of New Brunswick in Canada [26]. The NSL-KDD dataset is available in text-format (TXT-format) which consists of a train dataset, KDDTrain$^+$Attack, and a test dataset, KDDTest$^+$Attack, with labelled attacks (see Table 1 respectively Table 2). At their website there are also datasets available in ARFF-format, a train dataset, KDDTrain$^+$, and a test dataset, KDDTest$^+$, these datasets consists only of instances labelled as "normal" or "anomaly".

Weka requires the dataset in ARFF-format, consequently the datasets KDDTrain$^+$Attack and KDDTest$^+$Attack was pre-processed in Notepad++ 6.8 converting the data from TXT-format to ARFF-format.

Notepad ++ 6.8 was also used to modify the NSL-KDD into two new datasets where the labelled attacks in KDDTrain$^+$Attack and KDDTest$^+$Attack datasets are replaced by their corresponding attack category: Dos, Probe, U2R or R2L, according to Table 1 and Table 2. The modified datasets are henceforth referred to as KDDTrain$^+$Cat respectively KDDTest$^+$Cat. Table 6 below presents the datasets and the corresponding classes.

Table 6: Datasets with corresponding number of classes.

Dataset	Classes in dataset	#
KDDTrain$^+$ KDDTest$^+$	normal, anomaly	2
KDDTrain$^+$Cat KDDTest$^+$Cat	normal, dos, probe, u2r, r2l	5
KDDTrain$^+$Attack KDDTest$^+$Attack	normal, back, land, neptune, pod, smurf, teardrop, mailbomb, processTable, udpstorm, apache2, worm, satan, ipsweep, nmap, portsweep, mscan, saint, guess_passwd, ftp_write, imap, phf, multihop, warezmaster, warezclient, spy, xlock, xsnoop, snmpguess, snmpgetattack, httptunnel, sendmail, named, buffer_overflow, loadmodule, rootkit, perl. sqlattack, xterm, ps.	40

System used during experiments was running on Windows 8.1 Pro N x64 operating system with a Intel® Pentium® CPU G3220 3GHz processor and 8GB RAM.

The Weka workbench version 3.6.12 was used to compare selected data mining algorithms on KDDTrain$^+$Cat, KDDTest$^+$Cat and the cost matrix (see Table 4). The machine learning algorithms C4.5, CART, k-NN and Naïve Bayes are implemented in Weka with different names, see Table 7 below. All algorithms are evaluated once, with default settings in Weka.

Table 7: Machine learning algorithms and corresponding names in Weka.

Algorithm	Name in Weka
C4.5	Trees.J48
CART	Trees.SimpleCART
k-NN	Lazy.iBk
Naïve Bayes	Bayes.NaiveBayes

Selected performance measures for the data mining algorithms are accuracy, false-positive rate and average cost because this is objective and often used to evaluate performance of data mining algorithms. All performance measures are given from the output of Weka. To ensure that the algorithms which requires certain memory capacity can run in Weka, the heap size is increased to 4096MB. This was done by modifying the *maxheap* parameter in the Java properties file RunWeka.txt.

A comparison between the tree versions of the datasets in Table 6 was conducted to analyse if fewer attacks were detected depending on the number of classes in the dataset.

To evaluate the most important features in the dataset the filter method, presented in Chapter 2.2.6 about feature evaluation, was chosen due to the possibility to make an independent assessment based on general characteristics of the data. To rank features according to information gain is a method regularly used by researchers, therefore this method was elected.

Applying feature selection separately on the train and test dataset might result in a selection of different features. To analyse all attacks present in both training and testing in the DARPA 1998 dataset for intrusion detection evaluation the modified versions of NSL-KDD, KDDTrain$^+$Cat and KDDTest$^+$Cat, were combined into an additional dataset including all instances in both train and test datasets. The combined dataset is henceforth called KDDCat. This was done in Notepad++ 6.8.

The Weka Feature Evaluator was used to implement the information gain ranking filter to the KDDCat dataset. This gives a ranked list of all features according to information gain for all classes.
To achieve a ranked list according to information gain for *each* class the KDDCat dataset was pre-processed in Weka between each analysis. The pre-processing was done by applying an unsupervised filter to the dataset to remove instances of no interest with the filter *RemoveWithValues.* For example, when analysing instances in the DoS category the instances belonging to the classes Probe, R2L and U2R are removed. The remaining KDDCat then consists of instances in the classes Normal and DoS. Weka's Feature Evaluator was used to rank features according to information gain to distinguish between Normal and DoS attacks. This procedure was done for all attack categories.

3.6 Method discussion

Three measurement to ensure the study’s credibility is validity, reliability and objectivity. Many researchers argue that these three aspects must be considered in scientific studies. *Validity* represents the extent to which you actually measure what you intend to measure. *Reliability* represent the extent to which you can achieve the same results if the study is repeated. *Objectivity* represents the extent to which personal beliefs and values impacts the study. [27]

The study shall aim to achieve high validity, high reliability and high objectivity. The validity can be increased by considering various perspectives. The reliability can be increased by verifying the results several times. The objectivity can be increased by motivating the method choices to give the reader opportunity to consider the results of the study. [27]

3.6.1 Validity

The DARPA 1998 dataset is simulated network traffic and simulated cyber-attacks from 1998. This means that the data do not correspond to actual network traffic and real cyber-attacks. The performance measures for machine learning algorithms on this simulated dataset may not correspond to their performance on real network data and real cyber-attacks.

The DARPA 1998 dataset is almost ancient due to the fast development of technology since 1998 and the creativity of hackers launching strategic and advanced cyber-attacks. The validity of a dataset from 1998 can therefore be questioned in 2015, but it is also important to note that no comparable dataset is available online for researchers. The validity can possibly be increased by the fact that several researchers state that if methods for detecting intrusions and cyber-attacks don’t perform acceptable on DARPA 1998 then they will not perform well on real network traffic either.

The validity is increased by using the whole NSL-KDD dataset instead of sampling 10% the dataset as conducted in several related studies. According to Witten and Hall [3] a large training dataset provides better classification and a large test dataset provides an accurate error rate. The NSL-KDD dataset is free from reduced redundant records and therefore it is beneficial to use the whole dataset in research.

Additionally, an evaluation of possible difference between datasets with the same instances but with different number of classes is done to ensure validity for performance measures when using a dataset with the following classes: Normal, Dos, Probe, R2L and U2R.

3.6.2 Reliability

A high reliability is guaranteed due to the usage of whole NSL-KDD, all data is available online and therefore anyone can download and modify the dataset according to Chapter 3.5. The choice to compare the performance of machine learning algorithms with the NSL-KDD test dataset, KDDTest^{+}, instead of using cross-validation will increase the reliability because other researchers can achieve the same results if the study is repeated.

3.6.3 Objectivity

To increase objectivity the data mining algorithm was selected from the Top 10 list of data mining algorithms and the data mining software Weka was selected due to its wide use in research on data mining for intrusion detection. To increase the objectivity further all methodology choices have been motivated to give the reader opportunity to reflect on the results of this thesis.

4 Results

4.1 Comparison of machine learning algorithms

In Figure 1 and Figure 2 visualise the distribution of the train and test dataset according to the total number of instances in each of the categories: normal, Dos, Probe, R2L and U2R. The percentages have been rounded from the exact numbers of instances of each class presented in Appendix B and Appendix C. Figure 1 and Figure 2 shows that the distribution of the train and test dataset vary.

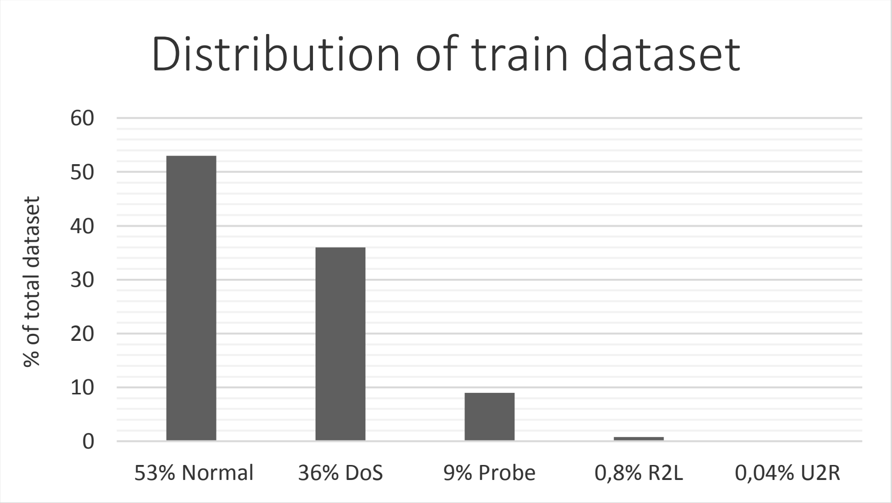

Figure 1: Distribution of train dataset, % of total train dataset KDDTrain+Cat.

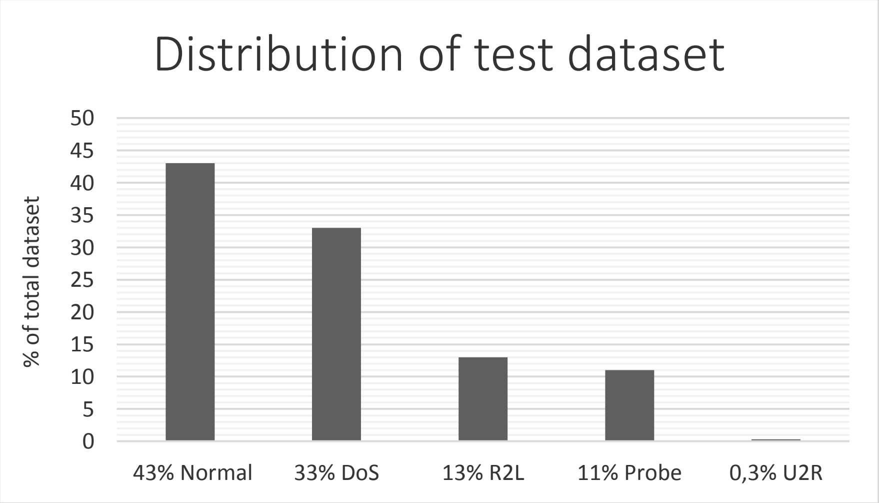

Figure 2: Distribution of test dataset, % of total test dataset KDDTest+Cat.

The comparison in Weka of the four machine learning algorithms are presented in Table 8 below. The performance measures are accuracy, error rate and average cost for classifying the test dataset KDDTest^{+}Cat.

Table 8: Output from Weka from KDDTest$^+$Cat for the algorithms k-NN, C4.5, CART and Naïve Bayes.

Algo-rithm	Name in Weka	Accuracy	Error rate	Cost
k-NN	Lazy.IBk	77.0892%	22.9108%	0.6612
C4.5	Trees.J48	75.2573%	24.7437%	0.6738
CART	Trees.SimpleCART	74.5697%	25.4303%	0.6647
Naïve Bayes	Bayes.NaiveBayes	71.203%	28.797%	0.6808

The results from Weka shows that k-Nearest Neighbour (k-NN) is the machine learning algorithm with best performance in all performance measures with an accuracy of 77.0892%, error rate of 22.9108% and the lowest average cost of 0.6612.

At second place the C4.5 algorithm have an accuracy of 75.2573%, error rate of 24.7437% and the average cost of 0.6738. At third place the CART algorithm have an accuracy of 74.5697%, error rate of 25.4303% and a lower average cost of 0.6647 compared to the C4.5. This indicates that the CART algorithm is better at classifying attacks with a high misclassification cost such as R2L attacks.

The algorithm with poorest performance on all performance measures is the Naïve Bayes with an accuracy of 71.203%, error rate of 28.797% and average cost of 0.6808.

4.2 Evaluation of classes used for classification

To evaluate if the performance of the machine learning algorithms depend on chosen number of classes, the train datasets and their corresponding testsets in Table 6 have been compared. KDDTrain$^+$ and KDDTest$^+$ have 2 classes, KDDTrain$^+$Cat and KDDTest$^+$Cat have 5 classes and KDDTrain$^+$Attack and KDDTest$^+$Attack have 40 classes. In Table 9 below the performance of the four machine learning algorithms are compared for the three datasets.

Table 9: Output from Weka for different datasets with the four machine learning algorithms.

Dataset	Algorithm	Accuracy	Error rate
KDDTest$^+$	C4.5	81.5%	18.5%
KDDTest$^+$Cat	C4.5	75.3%	24.7%
KDDTest$^+$Attack	C4.5	71.4%	28.6%

KDDTest$^+$	CART	80.3%	19.7%
KDDTest$^+$Cat	CART	74.6%	25.4%
KDDTest$^+$Attack	CART	61.3%	38.7%
KDDTest$^+$	kNN	79.4%	20.6%
KDDTest$^+$Cat	kNN	77.1%	22.9%
KDDTest$^+$Attack	kNN	70.3%	29.7%
KDDTest$^+$	Naive Bayes	76.1%	23.9%
KDDTest$^+$Cat	Naive Bayes	71.2%	28.8%
KDDTest$^+$Attack	Naive Bayes	22.4%	77.6%

The highest accuracy and lowest error rate is given by KDDTest$^+$ with the two classes normal and anomaly. This point out that all machine learning algorithms perform best when all attacks are in the class anomaly instead of divided into attack categories or specific attacks.

4.3 The most important features to detect cyber-attacks

The distribution in KDDCat which consists of KDDTrain$^+$Cat and-KDDTest$^+$Cat is presented in Figure 3 below.

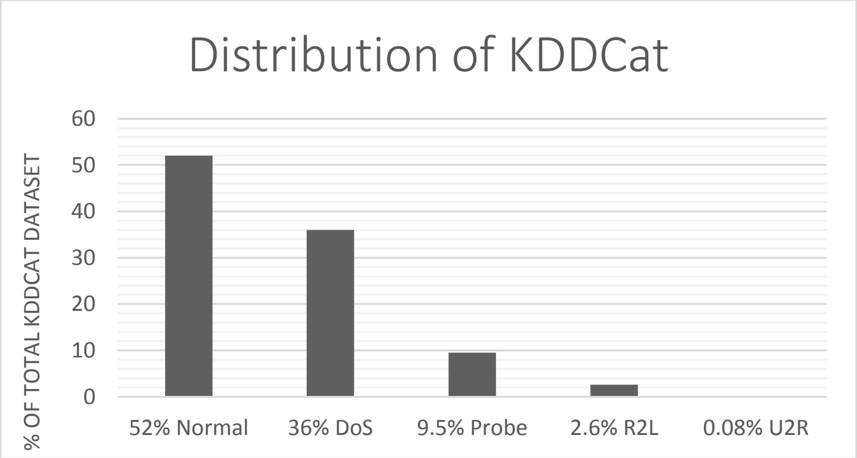

Figure 3: Distribution on KDDCat with rounded percentage.

The evaluation in Weka's Attribute Evaluator gives the result of which features that are most important, according to information gain, to distinguish between normal network traffic and cyber-attacks. The result is presented in Figure 4 below.

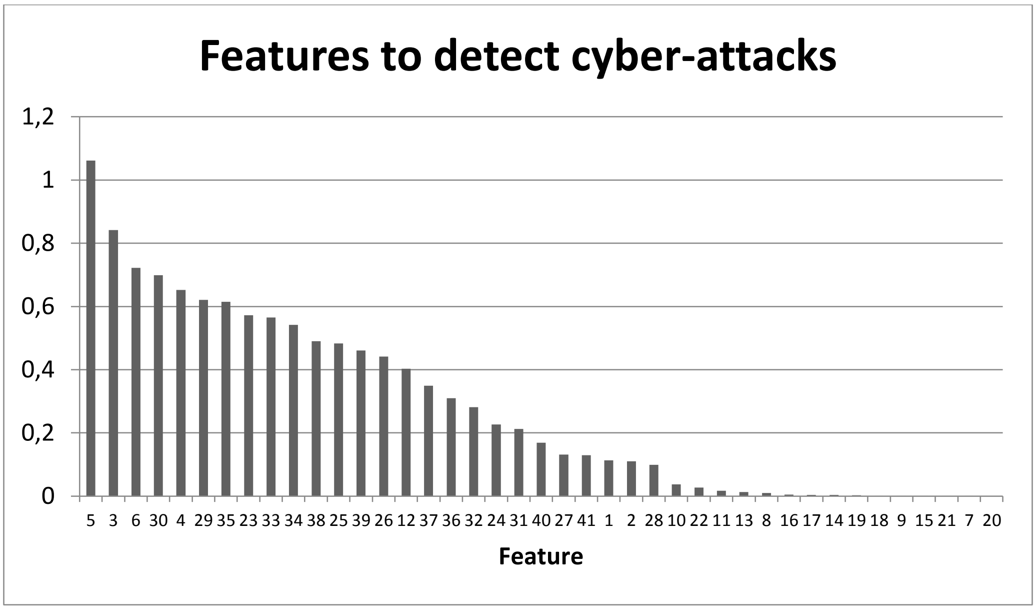

Figure 4: Ranked features according to Information Gain, for all classes in the KDDCat dataset.

The 5 most important features to detect any cyber-attacks are presented below in Table 10.

Table 10: Top 5 most important features to distinguish between normal network traffic and cyber-attacks.

Rank.	Feature Num.	Feature Name	Feature category
1.	5	Src_bytes	Basic Features
2.	3	Service	Basic Features
3.	6	Dst_bytes	Basic Features
4.	30	Diff_srv_rate	Time-based Traffic Features
5.	4	Flag	Basic Features

Table 10 shows that 4 of 5 features are from the category Basic Features and 1 of 5 features is from the category Time-based Traffic Features. The result from Weka shows that the Information Gain is zero for feature number 20 (num_outbound_cmds).

4.4 The most important features to detect DoS, Probing, R2L and U2R attacks.

The analysis of which features that are most important for *each* category of cyber-attacks are presented in Figure 5 for DoS, in Figure 6 for Probing, in Figure 7 for R2L and in Figure 8 for U2R. The feature's corresponding number is presented in Table 3 and all features are also described in Appendix A.

The result from computation of Information Gain in Weka is presented in Appendix E, Figure 4-8 is a visualisation of this result and Table 11-14 presents the five most important features. In Figure 5 below the features are ranked according to Information Gain.

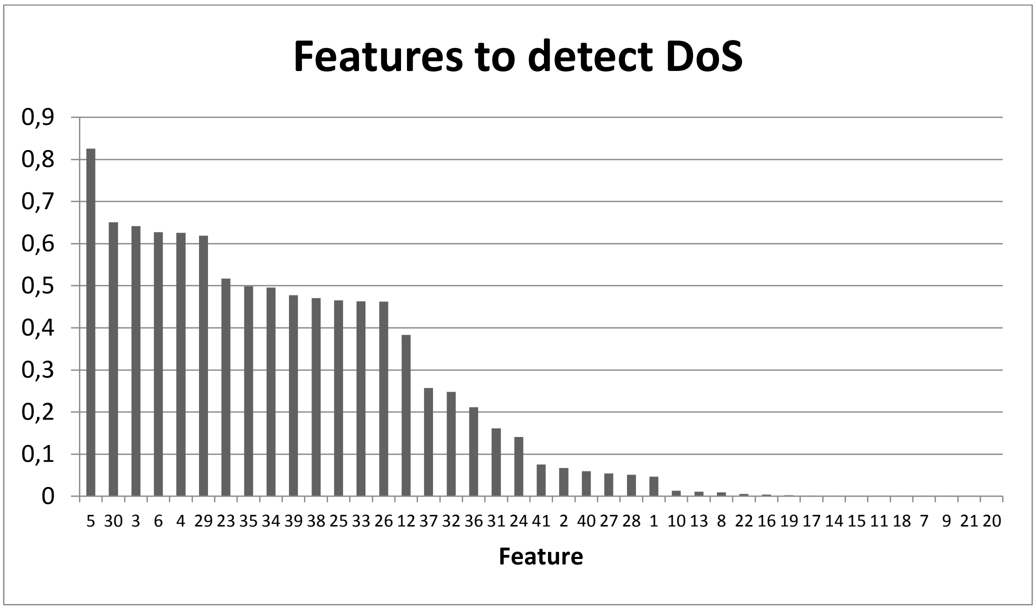

Figure 5: Ranked features according to Information Gain, to distinguish between normal traffic and DoS in the KDDCat dataset.

The 5 most important features to detect DoS attacks are presented below in Table 11.

Table 11: Top 5 most important features to distinguish between normal network traffic and DoS attacks.

Rank.	Feature Num.	Feature Name	Feature category
1.	5	Src_bytes	Basic Features
2.	30	Diff_srv_rate	Time-based Traffic Features
3.	3	Service	Basic Features
4.	6	Dst_bytes	Basic Features
5.	4	Flag	Basic Features

Table 11 shows that 4 of 5 most important features are from the category Basic Features and 1 of 5 features is from Time-based Traffic Features. The result from Weka shows that the Information Gain is zero for feature number 20 (num_outbound_cmds).

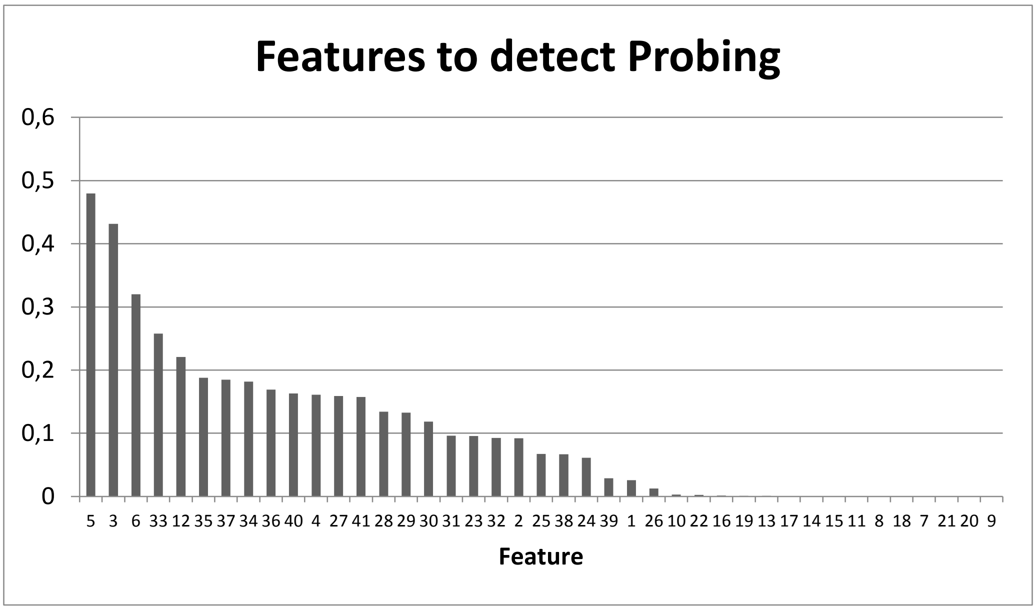

Figure 6: Ranked features according to Information Gain, to distinguish between normal traffic and Probing in the KDDCat dataset.

The 5 most important features to detect Probing attacks are presented below in Table 12.

Table 12: Top 5 most important features to distinguish between normal network traffic and Probing attacks.

Rank.	Feature Num.	Feature Name	Feature category
1.	5	Src_bytes	Basic Features
2.	3	Service	Basic Features
3.	6	Dst_bytes	Basic Features
4.	33	Dst_host_srv_count	Host-based Traffic Features
5.	12	Logged_in	Content Features

Table 12 shows that 3 of 5 most important features are from the category Basic Features, 1 of 5 features is from the category Host-based Traffic Features and 1 of 5 features is from the category Content Features. The result from Weka shows that the Information Gain are zero for feature number 20 (num_outbound_cmds) and feature number 9 (urgent).

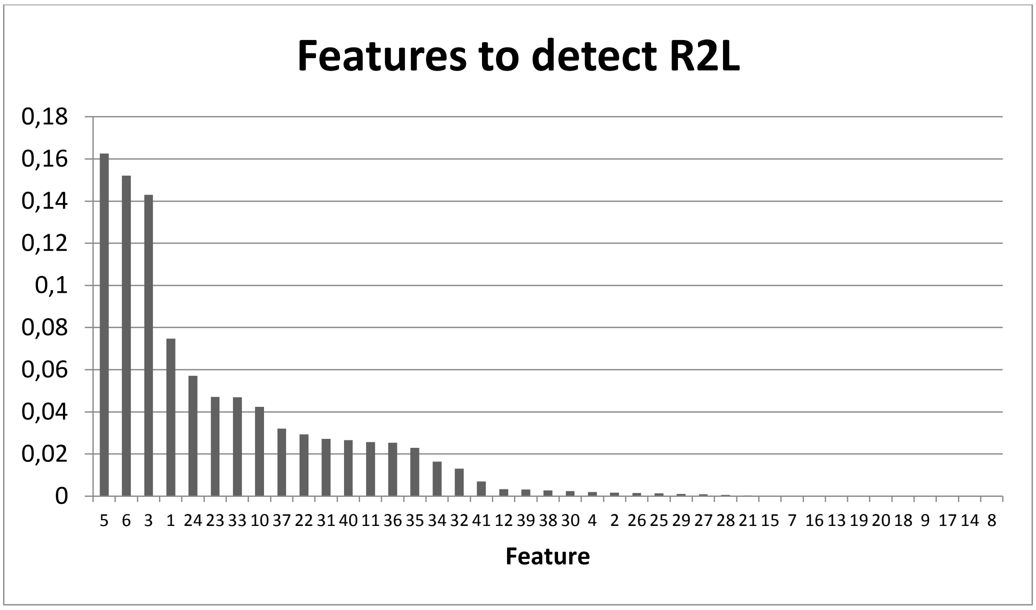

Figure 7: Ranked features according to Information Gain, to distinguish between normal traffic and R2L attacks in the KDDCat dataset.

The 5 most important features to detect R2L attacks are presented below in Table 13.

Table 13: Top 5 most important features to distinguish between normal network traffic and R2L attacks.

Rank.	Feature Num.	Feature Name	Feature category
1.	5	Src_bytes	Basic Features
2.	6	Dst_bytes	Basic Features
3.	3	Service	Basic Features
4.	1	Duration	Basic Features
5.	24	Srv_count	Time-based Traffic Features

Table 13 shows that 4 of 5 most important features are from the category Basic Features, 1 of 5 features is from the category Time-based traffic Features. The result from Weka shows that the Information Gain are zero for the following features: feature number 16 (num_root), feature 13 (num_compromised), feature 19 (num_access_files), feature 20 (num_outbound_cmds), feature 18(num_shells), feature 9 (urgent), feature 17 (num_file_creations), feature 14 (root_shell) and feature 8 (wrong_fragment).

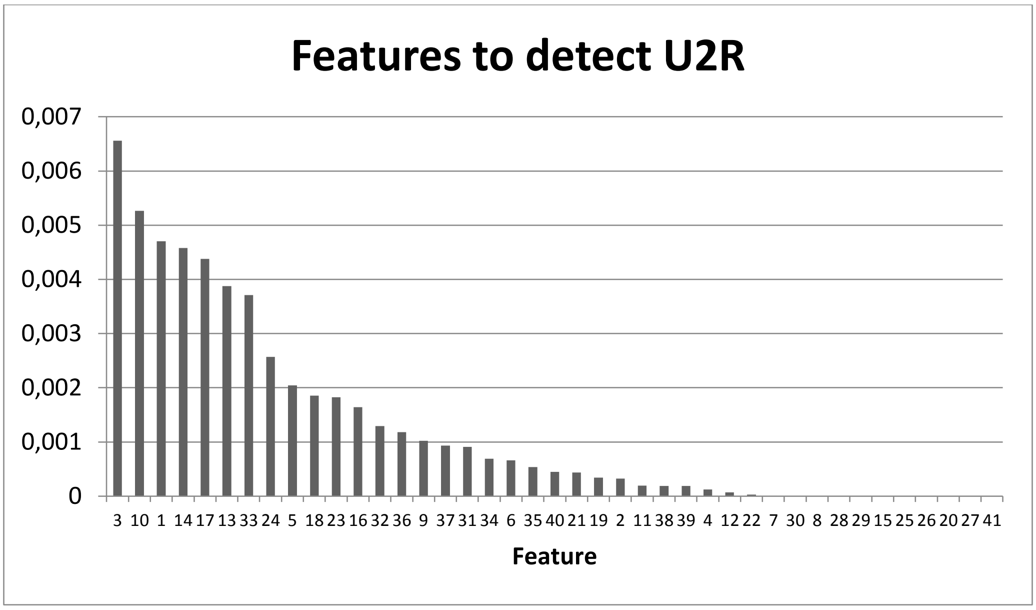

Figure 8: Ranked features according to Information Gain, to distinguish between normal traffic and U2R attacks in the KDDCat dataset.

The 5 most important features to detect U2R attacks are presented below in Table 14.

Table 14: Top 5 most important features to distinguish between normal network traffic and U2R attacks.

Rank.	Feature Num.	Feature Name	Feature category
1.	3	Service	Basic Features
2.	10	Hot	Content Features
3.	1	Duration	Basic Features
4.	14	Root_shell	Content Features
5.	17	Num_file_creations	Content Features

Table 14 shows that 2 of 5 most important features are from the category Basic Features, 3 of 5 features are from the category Content Features. The result from Weka shows that the Information Gain are zero for the following features: feature 30 (diff_srv_rate), feature 8 (wrong_fragment), feature 28 (srv_rerror_rate), feature 29 (same_srv_rate), feature 15 (su_attempted), feature 25 (serror_rate), feature 26 (srv_serror_rate), feature 20 (num_outbound_cmds), feature 27 (rerror_rate) and feature 41 (dst_host_srv_rerror_rate).

5 Discussion

5.1 Comparison of machine learning algorithms

By using the whole NSL-KDD dataset for comparing the four machine learning algorithms and not sampling 10% of the dataset, the algorithms have been given a large training set to ensure that the performance is optimal.

The distribution in train and test dataset presented in Figure 1 and Figure 2 shows the small proportion of R2L and U2R attacks. Obviously, it is a challenge for the machine learning algorithms to learn from the training data for these two categories of attacks. The cost matrix (Table 4) implies that the cost are greatest for the instances classified as normal when it is truly an R2L or U2R attack. Studying the average cost shows that the instance based machine learning algorithm k-NN is better of classifying R2L and U2R attacks.

The k-NN algorithm has the best performance in accuracy, error rate and average cost. A disadvantage with the k-NN algorithm commented by Witten and Hall in [3], is the large computation cost for all instance based learning algorithms. Therefore it is questionable how the k-NN algorithm can handle real network data and larger training sets, along with limited data capacity. For powerful systems the data capacity required to run k-NN is perhaps not an issue.

A possible explanation for the performance of the algorithms C4.5 and CART is overfitting. With 41 features in the datasets it is likely that the decision tree is not describing the underlying relationships and instead random error. According to Witten and Hall [3], overfitting can lead to good performance in training but poor performance on the test dataset which in this thesis includes novel attacks and a different distribution.

The performances of all four algorithms are evaluated in Weka using default settings instead of optimizing performance with suited settings for the dataset. The k-NN algorithm can be applied with a large value of *k* to decrease noise and misclassification, this gives better performance according to Witten and Hall [3]. If the dataset were pre-processed to fit each algorithm, the results would perhaps be different.

5.2 Evaluation of classes used for classification

The highest accuracy and lowest error rate are achieved for all machine learning algorithms when classifying the dataset with only the two classes normal and anomaly. The accuracy decreases and the error rate increases when the number of classes are extended. But without any information about which category of cyber-attack, it is impossible to know how to defend yourself. The cyber-attacks in DoS, Probing, R2L and U2R are very different and to know which cyber-attack category can give highly valuable information for security.

One reason for a lower accuracy and higher error rate for the datasets with more classes are that the machine learning algorithms misclassifies cyber-attacks as other cyber-attacks, this results in a lower accuracy and a higher error rate for classification.

The results point toward dividing the dataset into classes of normal, DoS, Probing, R2L and U2R to achieve a low error rate, high accuracy as well as knowledge about which category of cyber-attacks you are attacked with. To have 40 classes for classification provide mostly confusion and poor performance.

5.3 The most important features to detect cyber-attacks

Figure 3 shows that several features are important to detect cyber-attacks. The category with least importance for detecting cyber-attacks is the Content Based Features. This result can be biased because of the distribution of train and test datasets where the U2R and R2L attacks is less frequent than the other categories of cyber-attacks. The most important features for U2R and R2L attacks consequently have a smaller information gain than the most important features for DoS and Probing attacks.

The results show that the Basic Features are most important to analyse to distinguish between normal network traffic and cyber-attacks. The most important features from this category are feature number 1, 2, 3, 4, 5 and 6. These features describes the number of seconds for the connection, the protocol used for the connection, the network service on the destination, normal or error status of the connection and the number of data bytes sent between source and destination computer. The results indicate the significance of analysing basic network traffic features to detect cyber-attacks.

The category with the least important features is the Content Features, this result is very interesting thus it implies that it is maybe not necessary to analyse contents in the network traffic packages to detect cyber-attacks. The aspect of privacy and integrity for employees is then protected. A content feature can be the text in an email, and to store and analyse this kind of content violates the employees' integrity.

In Table 10 the most important features are presented. The feature number 5, src_bytes, is important analyse to detect cyber-attacks since the value of feature 5 is the size (in data bytes) of the package sent from the source to the destination computer. This feature is unusually small or unusually large for cyber-attacks compared to normal network connections. Feature number 6, dst_bytes, is important to detect cyber-attacks for the same reason. Feature 6 is the size (in data bytes) of the package sent from the destination to the source computer.

Feature number 3, service, is important according to Table 10. This feature is important due to the difference between normal network connections and cyber-attacks. Several cyber-attacks in different categories exploit otherwise uncommon services. Feature number 3 can therefore indicate suspicious behaviour.

The Time-based Traffic Features consists of 100 connections each and the most important Time-based Feature to detect cyber-attacks is number 30, diff_srv_rate. The value of Feature 30 is the percentage of the connections in the past 2 seconds using a different service than the current connection. The basic feature number 4 is also important to detect cyber-attacks as stated by Table 10. Feature number 4, flag, indicates if a connection is incorrectly cancelled.

5.4 The most important features to detect DoS, Probing, R2L and U2R attacks.

The five most important attributes to detect DoS attacks are the same as presented in Chapter 5.3, feature number 5, 30, 3, 6 and 4. The ranking of these five features are not the same for DoS attacks, the time-based feature number 30 have a higher ranking for DoS attacks. The reason is that all DoS attacks, except Neptune, will have a low value for feature 30 compared to normal network traffic due to the attackers intention to overwhelm a specific service.

The content features are the least important category of features to detect a DoS attack. The reason for this is that the DoS attacks mostly consist of either no content or filled with a large amount of useless information.

Some cyber-attacks are ongoing for a longer time than 2 seconds and in multiple connections. This is the reason for the Host-based feature number 33, dst_host_srv_count, is important to detect probing attacks since the rate of changing service will be high when an attacker is probing different ports. The probing attacks usually seek known vulnerabilities, therefore feature number 12, logged_in, representing a successful login attempt is important for detecting probing attacks.

For R2L attacks, the basic feature number 1, duration, is one of the most important features besides the already mentioned features number 5, 6 and 3. The importance of analysing feature number 1 is that the value of this feature is the number of seconds for the connection. Several R2L attacks have a duration which is much larger than a normal connections.

Another important feature to detect R2L attacks is the time-based feature number 24, srv_count, which represent the number of connections in the past 2 seconds to the same service as the current connection. During a R2L attack, the attacker tries to gain access to a local user account with a specific service in connections longer than 2 seconds. For R2L attacks, feature number 24 will therefore have a low value compared to normal network traffic.

To detect U2R attacks the most important feature is the basic feature number 3, service, since the U2R attacks involves the use of specific services for remote access, often in combination with a file transfer service. Compared to the other attack categories, the content features are very important to detect U2R attacks. The content features are created by analysing the content in a network connection. The importance of content features to detect U2R attacks are because of the remote users actions can only be noticed when analysing the content in the connection packages.

6 Conclusions

This thesis most important contribution is the expansion of knowledge about data mining for network intrusion detection in order to detect cyber-attacks and determine defence configuration.

This thesis shows that k-Nearest Neighbour is the most accurate classification machine learning algorithm, from the Top 10 data mining algorithms presented by [5] and [3], for detection of cyber-attacks in the NSL-KDD dataset.

The results display the importance of choosing a relevant number of classes for classification. Although it is interesting to know what specific cyber-attack you are exposed to, the large number of classes for classification decreases the accuracy of all the machine learning algorithms. To increase the possibility to make well-founded decisions about defence configuration, the best classes for classification are categories of cyber-attacks such as DoS, Probing, R2L and U2R attacks. This vide categorization gives the opportunity to detect novel attacks and increases the knowledge about how to encounter cyber-attacks.

The most important features to detect cyber-attacks are basic features such as information about the size of the package, the used service, a flag to indicate the status of the connection. Moreover time-based traffic features is important to analyse to detect cyber-attacks, such as information about the percentage of connections in the past 2 seconds with a different service than current connection. To detect R2L and U2R attacks it is important to study content features.

6.1 Ethical aspects

Methods to detect threats, intrusions and suspicious behaviour such as data mining can be criticised regarding ethical aspects. The use of data mining can affect the integrity by logging for instance the employees' arrival, internet habits and interests. This is a balance between security and surveillance. Two important questions to consider are: When does data mining violate privacy? How can data mining be misused to generate information about legitimate users?

6.2 Future work

A suggestion for future work is to compare machine learning algorithms on real network traffic in different environments, to explore if the algorithms perform better or worse with changed conditions.

An interesting topic for further research is to create an updated dataset with normal network traffic similar to a system in 2015 and modern attacks such as Cross-site scripting (XSS) attacks and Structured Query Language injections (SQLi) attacks. An updated dataset can give more information about present conditions that needs to be considered.

The results in this thesis shows that content features from network traffic are important to detect R2L and U2R attacks. Further research to determine if you can detect attacks in these two categories without analysing content features is necessary.

Future work with great relevance is to study risk assessment regarding network intrusion detection. It is important to have knowledge about how to counteract cyber-attacks to ensure security. If a cyber-attack is detected what should be done to interrupt or cancel? What can be done to protect your assets?

References

[1] Andersson, Ross. 2008. *Security engineering: A guide to Building Dependable Distributed Systems.* 2nd edition. U.S: John Wiley & Sons.

[2] De Ocampo, Frances Bemadette C.; Del Castillo, Trisha Mari L.; Gomez, Miguel Alberto N. 2013. *Automated signature creator for a signature based intrusion detection system with network attack detection capabilities (pancakes).* International Journal of Cyber-Security and Digital Forensics, Jan, 2013, Vol.2(1).

[3] Witten, Ian H.; Frank, Eibe; Hall, Mark A. 2011. *Data mining – Practical Machine Learning Tools and Techniques.* Third Edition. USA: Elsevier Inc.

[4] Lincoln Laboratory, Massachusetts Institute of technology. 2014. *Cyber Systems and Technology – DARPA Intrusion Detection Data Sets – 1998 dataset.* URL http://www.ll.mit.edu/ideval/data/. Retrieved 2015-06-25.

[5] Wu, Xindong; Kumar, Vipin; Ross Quinlan, J.; Ghosh, Joydeep; Yang, Qiang; Motoda, Hiroshi; McLachlan, Geoffrey; Ng, Angus; Liu, Bing; Yu, Philip; Zhou, Zhi-Hua; Steinbach, Michael; Hand, David; Steinberg, Dan. 2008. *Top 10 algorithms in data mining.* Knowledge and Information Systems, 2008, Vol.14.

[6] Waxman, Matthew C. 2011. *Cyber-Attacks and the Use of Force'.* Yale Journal of International Law. Vol. 36.

[7] Thomas, Ciza; Sharma, Vishwas; Balakrishnan, N. Dasarathy, Belur V. 2008. *Usefulness of DARPA dataset for intrusion detection system evaluation.* Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008, Monday 17 March 2008, Vol.6973(1).

[8] Tavallaee, Mahbod; Bagheri, Ebrahim; Lu, Wei; Ghorbani, Ali A. 2009. *A detailed analysis of the KDD CUP 99 data set.* 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, July 2009, pp.1-6.

[9] Saydjari, O., Sami. 2004. *Cyber defense: art to science*. Communications of the ACM, March 2004, Vol.47(3).

[10] Fayyad, Usama; Piatetsky Shapiro, Gregory; Smyth, Padhraic. 1996. *From data mining to knowledge discovery in databases.* AI Magazine, Fall, Vol.17(3).

[11] H. G. Kayacık, A. N. Zincir-Heywood, M. I. Heywood. 2005. *Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets.* Canada, Nova Scotia, Halifax, Dalhousie University. Faculty of Computer Science.

[12] Japkowicz, Nathalie; Shah, Mohak. 2011. *Evaluating Learning Algorithms : A Classification Perspective.* USA: Cambridge University Press.

[13] The University of Waikato. 2015. Weka 3: Data mining software in Java. URL http://www.cs.waikato.ac.nz/ml/weka/. Retrieved 2015-05-29.

[14] R. P. Lippmann; D. J. Fried; I. Graf; J. W. Haines; K. R. Kendall, D. McClung; D. Weber; S. E. Webster; D. Wyschogrod; R. K. Cunningham; M. A. Zissman. 2000. *Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation.* DARPA Information Survivability Conference and Exposition (DISCEX) 2000, vol.2.

[15] S.J. Stolfo, W. Fan, W. Lee. 2000. *Cost-Based modeling and evaluation for data mining with application to fraud and intrusion detection: Results from the JAM project*. Proceedings of the DARPA Information Survivability Conference & Exposition.

[16] McHugh, John. 2000. *Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA IDS evaluations as performed by Lincoln Laboratory.* ACM Transactions on Information and System Security, vol.3, No.4. November 2000, p. 262-294.

[17] M. V. Mahoney, P. K. Chan. 2003. *An analysis of the 1999 DARPA /Lincoln Laboratory evaluation data for network anomaly detection*. Technical Report CS-2003-02.

[18] Elkan, Charles, *Results of the KDD'99 classifier learning*, ACM SIGKDD Explorations Newsletter, January 2000, Vol.1(2).

[19] Naahid, Shams; Siddiqui, Mohammad Khubeb. 2013. *Analysis of KDD CUP 99 Dataset using Clustering based Data Mining.* International Journal of Database Theory and Application. Vol.6, No.5.

[20] Mittal, Saurabh; Richa. 2014. *Data Mining Approach IDS K-Mean using Weka Environment.* International Journal of Advanced Research in Computer Science and Software Engineering. Vol.4, No.8.

[21] Kailashiya, Devendra; Jain, R.C. 2012. *Improve Intrusion Detection Using Decision Tree with Sampling.* International Journal of Computer Technology & Applications. Vol. 3, No.3.

[22] Chandolikar, N.S; Nandavadekar, V.D. 2012. *Selection of Relevant Feature for Intrusion Attack Classification by Analyzing KDD Cup 99.* MIT International Journal of Computer Science & Information Technology, Vol. 2, No. 2.

[23] Kulkarni, Rohan D. 2014. *Using Ensemble Methods for Improving Classification of the KDD CUP '99 Data Set.* IOSR Journal of Computer Engineering.Vol.16, No. 5.

[24] Dartigue, Christine; Jang, Hyun Ik; Zeng, Wenjun. 2009. *A New Data-Mining Based Approach for Network Intrusion Detection.* 2009 Seventh Annual Communication Networks and Services Research Conference, May 2009, pp.372-377.

[25] Ajayi Adebowale, Idowu S.A, Anyaehie Amarachi A. 2013. *Comparative study of selected data mining algorithms used for intrusion detection.* International Journal of Soft Computing and Engineering (IJSCE), Vol.3(3).

[26] The University of Waikato. 2015. Weka 3: Data mining software in Java. URL http://www.cs.waikato.ac.nz/ml/weka/. Retrieved 2015-05-29.

[27] Björklund, Maria; Paulsson, Ulf. 2012. *Seminarieboken: att skriva, presentera och opponera.* Edition 2:3. Lund: Studentlitteratur.

Appendix A: Features in DARPA 1998 Dataset

Description of basic features, content features, traffic features and host-based features.

Table A1: Basic Features. [15]

No.	Feature	Description	Type
1.	Duration	Length (number of seconds) of the connection.	Cont.
2.	Protocol_Type	Type of the protocol, e.g. tcp, udp, etc.	Disc.
3.	Service	Network service on the destination, e.g., http, telnet, etc.	Disc.
4.	Flag	Normal or error status of the connection.	Disc.
5.	Src_Bytes	Number of data bytes from source to destination.	Cont.
6.	Dst_Bytes	Number of data bytes from destination to source.	Cont.
7.	Land	1 - Connection is from/to the same host/port; 0 – otherwise.	Disc.
8.	Wrong_Fragment	Number of “wrong” fragments.	Cont.
9.	Urgent	Number of urgent packets.	Cont.

Table A2: Content Features. [15]

No.	Feature	Description	Type
10.	Hot	number of “hot indicators”.	Cont.
11.	num_failed_logins	number of failed login attempts	Cont.
12.	logged_in	1 - successfully logged in; 0 - otherwise	Disc.
13.	num_compromised	number of “compromised” conditions.	Cont.
14.	root_shell	1 - root shell is obtained; 0 – otherwise.	Cont.
15.	su_attempted	1 - “su root” command attempted; 0 – otherwise.	Cont.
16.	num_root	number of “root” accesses.	Cont.

17.	num_file_creations	number file creation operations	Cont.
18.	num_shells	number of shell prompts	Cont.
19.	Num_access_files	number of operations on access control files	Cont.
20.	Num_out-bound_cmds	number of outbound commands in a ftp session	Cont.
21.	is_hot_login	1 - the login belongs to the "hot" list; 0 – otherwise.	Disc.
22.	is_guest_login	1 - the login is a "guest"login; 0 - otherwise	Disc.

Table A3: Time-based Traffic Features. [15]

No.	Feature	Description	Type
23.	Count	number of connections to the same host as the current connection in the past 2 seconds	Cont.
the following features refer to these same-host connections			
24.	srv_count	number of connections to the same service as the current connection in the past 2 seconds	Cont.
25.	serror_rate	% of connections that have "SYN" errors	Cont.
27.	rerror_rate	% of connections that have "REJ" errors	Cont.
29.	same_srv_rate	% of connections to the same service	Cont.
30.	diff_srv_rate	% of connections to different services	Cont.
the following features refer to these same-service connections			
26	srv_serror_rate	% of connections that have "SYN" errors	Cont.
28	srv_rerror_rate	% of connections that have "REJ" errors	Cont.

Table A4: Host-based Traffic Features. [11]

No.	Feature	Description	Type
31	srv_diff_host_rate	% of connections to different hosts	Cont.
32	dst_host_count	count of connections having the same destination host	Cont.
33	dst_host_srv_count	count of connections having the same destination host and using the same service	Cont.
34	dst_host_same_srv_ rate	% of connections having the same destination host and using the same service	Cont.
35	dst_host_diff_srv_rate	% of different services on the current host	Cont.
36	dst_host_same_src_port_rate	% of connections to the current host having the same src port	Cont.
37	dst_host_srv_diff_host_rate	% of connections to the same service coming from different hosts	Cont.
38	dst_host_serror_rate	% of connections to the current host that have an S0 error	Cont.
39	dst_host_srv_serror_rate	% of connections to the current host and specified service that have an S0 error	Cont.
40	dst_host_rerror_rate	% of connections to the current host that have an RST error	Cont.
41	dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an RST error	Cont.

Appendix B: Distribution of train dataset

Table B1: Distribution of train dataset; quantity of each attack in train dataset.

#	Attack	#	Attack
956	back	3	perl
30	buffer_overflow	4	phf
8	ftp_write	201	pod
53	guess_passwd	2931	portsweep
11	imap	10	rootkit
3599	ipsweep	3633	satan
18	land	2646	smurf
9	loadmodule	2	spy
7	multihop	892	teardrop
41214	neptune	890	warezclient
1493	nmap	20	warezmaster
67343	normal		

Table B2: Distribution of train dataset; quantity and percentage of each attack category.

Class		Number of instances	% of total dataset
DoS	956+18+41214+201+2646+892	45927	36%
Probe	3633+3599+1493+2931	11656	9%
R2L	53+8+11+4+7+890+20+2	995	0,8%
U2R	30+9+10+3	52	0,04%
Normal		67343	53%
Sum		125 973	

Appendix C: Distribution of test dataset

Table C1: Distribution of test dataset; quantity of each attack in test dataset.

#	**Attack**	#	**Attack**
359	back	41	pod
20	buffer_overflow	157	portsweep
3	ftp_write	13	rootkit
2131	guess_passwd	735	satan
1	imap	665	smurf
141	ipsweep	12	teardrop
7	land	944	warezmaster
2	loadmodule	319	saint
18	multihop	996	mscan
4657	neptune	737	apache2
73	nmap	178	snmpgetattack
9711	normal	685	processTable
2	perl	133	httptunnel
2	phf	15	ps
331	snmpguess	293	mailbomb
17	named	14	sendmail
13	xterm	2	worm
9	xlock	4	xsnoop
2	sqlattack	2	udpstorm

Table C2: Distribution of test dataset; quantity and percentage of each attack category.

Class		**Number of instances**	**% of total dataset**
DoS	359+7+4657+41+665+12+293+685+2+737+2	7460	33%
Probe	735+141+73+157+996+319	2421	11%
R2L	1231+3+1+2+18+944+9+4+331+178+133+14+17	2885	13%
U2R	20+2+13+2+2+13+15	67	0,3%
Normal		9711	43%
total		22544	

Appendix D: Distribution of KDDCat dataset

Table D1: Number of each attack in KDDCat dataset.

#	**Attack**	#	**Attack**
1315	back	242	pod
50	buffer_overflow	3088	portsweep
11	ftp_write	23	rootkit
2184	guess_passwd	4368	satan
12	imap	3311	smurf
3740	ipsweep	904	teardrop
25	land	964	warezmaster
11	loadmodule	890	warezclient
25	multihop	319	saint
45871	neptune	996	mscan
1566	nmap	737	apache2
77054	normal	178	snmpgetattack
5	perl	685	processTable
6	phf	133	httptunnel
331	snmpguess	15	ps
17	named	293	mailbomb
13	xterm	14	sendmail
9	xlock	2	worm
2	sqlattack	4	xsnoop
2	spy	2	udpstorm

Table D2: Distribution of KDDCat dataset.

Class	**Number of instances**	**Percentage of total dataset**
normal	77054	52%
dos	53387	36%
probe	14077	9,5%
R2l	3880	2,6%
U2r	119	0,08%
total		148517

Appendix E: Information Gain for Features in KDDCat

Information Gain for features to distinguish between normal network traffic and cyber-attacks.

Ranked attributes:

1.061216	5	src_bytes
0.841472	3	service
0.72251	6	dst_bytes
0.698994	30	diff_srv_rate
0.652719	4	flag
0.620714	29	same_srv_rate
0.615178	35	dst_host_diff_srv_rate
0.5722	23	count
0.564747	33	dst_host_srv_count
0.542068	34	dst_host_same_srv_rate
0.490396	38	dst_host_serror_rate
0.483369	25	serror_rate
0.460524	39	dst_host_srv_serror_rate
0.441983	26	srv_serror_rate
0.402938	12	logged_in
0.349294	37	dst_host_srv_diff_host_rate
0.309777	36	dst_host_same_src_port_rate
0.281277	32	dst_host_count
0.227315	24	srv_count
0.213051	31	srv_diff_host_rate
0.168721	40	dst_host_rerror_rate
0.131812	27	rerror_rate
0.130018	41	dst_host_srv_rerror_rate
0.11346	1	duration
0.110587	2	protocol_type
0.098959	28	srv_rerror_rate
0.037498	10	hot
0.026776	22	is_guest_login
0.017499	11	num_failed_logins
0.013015	13	num_compromised
0.009639	8	wrong_fragment
0.004814	16	num_root

```
0.004091   17 num_file_creations
0.003707   14 root_shell
0.002791   19 num_access_files
0.001362   18 num_shells
0.000658    9 urgent
0.000528   15 su_attempted
0.000428   21 is_host_login
0.00013     7 land
0         20 num_outbound_cmds
```

Information Gain for features to distinguish between normal network traffic and DoS attacks

```
Ranked attributes:
 0.82538491    5 src_bytes
 0.65093538   30 diff_srv_rate
 0.64175469    3 service
 0.62715666    6 dst_bytes
 0.62586769    4 flag
 0.6187154    29 same_srv_rate
 0.51676294   23 count
 0.49852165   35 dst_host_diff_srv_rate
 0.49589203   34 dst_host_same_srv_rate
 0.47761248   39 dst_host_srv_serror_rate
 0.47039092   38 dst_host_serror_rate
 0.46546048   25 serror_rate
 0.46277731   33 dst_host_srv_count
 0.4619961    26 srv_serror_rate
 0.38322878   12 logged_in
 0.2572653    37 dst_host_srv_diff_host_rate
 0.24781159   32 dst_host_count
 0.2117626    36 dst_host_same_src_port_rate
 0.16163101   31 srv_diff_host_rate
 0.14083979   24 srv_count
 0.07540427   41 dst_host_srv_rerror_rate
 0.06729167    2 protocol_type
 0.05930886   40 dst_host_rerror_rate
 0.05385794   27 rerror_rate
 0.05137516   28 srv_rerror_rate
 0.04659663    1 duration
 0.01328176   10 hot
```

```
0.01067105   13 num_compromised
0.00925919    8 wrong_fragment
0.00536383   22 is_guest_login
0.00372072   16 num_root
0.0024096    19 num_access_files
0.00150993   17 num_file_creations
0.00090315   14 root_shell
0.00047762   15 su_attempted
0.00042518   11 num_failed_logins
0.00022711   18 num_shells
0.00010187    7 land
0.00003493    9 urgent
0.00000582   21 is_host_login
0            20 num_outbound_cmds
```

Information Gain for features to distinguish between normal network traffic and Probing attacks

Ranked attributes:

```
 0.47967425    5 src_bytes
 0.43164867    3 service
 0.32007877    6 dst_bytes
 0.25764699   33 dst_host_srv_count
 0.22081248   12 logged_in
 0.18779043   35 dst_host_diff_srv_rate
 0.18461185   37 dst_host_srv_diff_host_rate
 0.18176      34 dst_host_same_srv_rate
 0.16922468   36 dst_host_same_src_port_rate
 0.16315097   40 dst_host_rerror_rate
 0.16082905    4 flag
 0.15902471   27 rerror_rate
 0.15765329   41 dst_host_srv_rerror_rate
 0.13412058   28 srv_rerror_rate
 0.13252617   29 same_srv_rate
 0.11859414   30 diff_srv_rate
 0.09623446   31 srv_diff_host_rate
 0.09545956   23 count
 0.09266645   32 dst_host_count
 0.09185482    2 protocol_type
 0.06701332   25 serror_rate
 0.06667335   38 dst_host_serror_rate
```

```
0.0609657    24 srv_count
0.02870491   39 dst_host_srv_serror_rate
0.02578393    1 duration
0.01265344   26 srv_serror_rate
0.00297198   10 hot
0.00235486   22 is_guest_login
0.00143177   16 num_root
0.00109977   19 num_access_files
0.00080273   13 num_compromised
0.00056232   17 num_file_creations
0.00041211   14 root_shell
0.00021792   15 su_attempted
0.00013969   11 num_failed_logins
0.00013817    8 wrong_fragment
0.00010362   18 num_shells
0.00001859    7 land
0.00000266   21 is_host_login
0          20 num_outbound_cmds
0           9 urgent
```

Information Gain for features to distinguish between normal network traffic and R2L attacks

```
Ranked attributes:
 0.16251906    5 src_bytes
 0.15215265    6 dst_bytes
 0.14294471    3 service
 0.07470952    1 duration
 0.05710336   24 srv_count
 0.04704098   23 count
 0.04699163   33 dst_host_srv_count
 0.04240363   10 hot
 0.03205154   37 dst_host_srv_diff_host_rate
 0.02930487   22 is_guest_login
 0.02714802   31 srv_diff_host_rate
 0.02654788   40 dst_host_rerror_rate
 0.02560263   11 num_failed_logins
 0.02530997   36 dst_host_same_src_port_rate
 0.02298459   35 dst_host_diff_srv_rate
 0.01643802   34 dst_host_same_srv_rate
 0.01303675   32 dst_host_count
```

```
0.00694382   41 dst_host_srv_rerror_rate
0.00336763   12 logged_in
0.00322831   39 dst_host_srv_serror_rate
0.00265567   38 dst_host_serror_rate
0.00236755   30 diff_srv_rate
0.00196084    4 flag
0.00168378    2 protocol_type
0.0014593    26 srv_serror_rate
0.00132302   25 serror_rate
0.00098109   29 same_srv_rate
0.00086475   27 rerror_rate
0.00061452   28 srv_rerror_rate
0.00032631   21 is_host_login
0.00005344   15 su_attempted
0.00000613    7 land
0            16 num_root
0            13 num_compromised
0            19 num_access_files
0            20 num_outbound_cmds
0            18 num_shells
0             9 urgent
0            17 num_file_creations
0            14 root_shell
0             8 wrong_fragment
```

Information Gain for features to distinguish between normal network traffic and U2R attacks

```
Ranked attributes:
 0.006557542    3 service
 0.005267563   10 hot
 0.004705531    1 duration
 0.004580571   14 root_shell
 0.004381022   17 num_file_creations
 0.003878987   13 num_compromised
 0.003713158   33 dst_host_srv_count
 0.002572772   24 srv_count
 0.002043733    5 src_bytes
 0.00185337    18 num_shells
 0.001827016   23 count
 0.001643437   16 num_root
 0.001295476   32 dst_host_count
```

0.00117962	36	dst_host_same_src_port_rate
0.001020748	9	urgent
0.000933856	37	dst_host_srv_diff_host_rate
0.000907611	31	srv_diff_host_rate
0.000689585	34	dst_host_same_srv_rate
0.000658755	6	dst_bytes
0.000534722	35	dst_host_diff_srv_rate
0.000446977	40	dst_host_rerror_rate
0.000438682	21	is_host_login
0.000344635	19	num_access_files
0.000323678	2	protocol_type
0.000193509	11	num_failed_logins
0.000191106	38	dst_host_serror_rate
0.000190055	39	dst_host_srv_serror_rate
0.000120597	4	flag
0.000068889	12	logged_in
0.000026613	22	is_guest_login
0.000000202	7	land
0	30	diff_srv_rate
0	8	wrong_fragment
0	28	srv_rerror_rate
0	29	same_srv_rate
0	15	su_attempted
0	25	serror_rate
0	26	srv_serror_rate
0	20	num_outbound_cmds
0	27	rerror_rate
0	41	dst_host_srv_rerror_rate